

The image is a large, symmetrical, abstract graphic composed of the letters 'S' and 'Y' arranged in a grid-like pattern. The overall shape is a stylized 'Y' or a complex letterform. The top part is a wide horizontal bar made of 'S's, with 'Y's forming a central vertical stem. The sides of the 'Y' are also formed by 'S's and 'Y's, creating a sense of depth and structure. The letters are black on a white background, and the arrangement is highly regular and repetitive, suggesting a digital or algorithmic origin.

IF  
VC

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

(2)	87	USEFUL MACRO DEFINITIONS
(3)	102	SYMBOL DEFINITIONS
(4)	165	ERROR MESSAGES AND MISC STRINGS
(4)	184	CHARACTER VALIDITY TABLES
(5)	195	COMMAND INTERPRETER AND DISPATCHER
(6)	276	CANCEL A PENDING MOUNT VERIFICATION
(7)	337	COMMAND LINE INPUT ROUTINE
(8)	425	MISCELLANEOUS CONSOLE OUTPUT ROUTINES



```
0000 1      .TITLE  IPCONTROL - IPL 12 INTERRUPT PROCESSOR
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10
0000 11      *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12      *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13      *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14      *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15      *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16      *  TRANSFERRED.
0000 17
0000 18      *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19      *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20      *  CORPORATION.
0000 21
0000 22      *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23      *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28 ++
0000 29
0000 30 FACILITY:
0000 31
0000 32      NONPAGED VAX/VMS EXEC
0000 33
0000 34 ABSTRACT:
0000 35
0000 36      THIS MODULE CONTAINS THE ROUTINE NECESSARY TO PROCESS THE IPL 12
0000 37      INTERRUPTS. THESE ARE GENERATED MANUALLY FROM THE CONSOLE AS A
0000 38      LAST DITCH EFFORT TO CORRECT SOME SYSTEM MALADY BEFORE REBOOTING
0000 39      SYSTEM.
0000 40
0000 41 AUTHOR:
0000 42
0000 43      L. MARK PILANT 28-JAN-1982
0000 44
0000 45 MODIFIED BY:
0000 46
0000 47      V03-010 LMP0294      L. Mark Pilant,      2-Aug-1984 15:54
0000 48      Add the dollar sign to the list of legal characters.
0000 49
0000 50      V03-009 PRB0333      Paul Beck      3-May-1984 22:26
0000 51      Fix fork block initialization for Q command.
0000 52
0000 53      V03-008 ACG0416      Andrew C. Goldstein, 18-Apr-1984 9:51
0000 54      Use common I/O search routines to support cluster devices
0000 55
0000 56      V03-007 PRB0328      Paul Beck      9-Apr-1984 15:59
0000 57      Add retry logic for call to CNX$CHANGE_QUORUM.
```

0000	58	:	
0000	59	:	
0000	60	:	V03-006 TCM0003 Trudy C. Matthews 09-Apr-1984
0000	61	:	Yet another change in RXCS/TXCS handling: use routine
0000	62	:	CONSRELEASECTY to restore the state of the registers.
0000	63	:	
0000	64	:	V03-005 PRB0325 Paul Beck 25-Mar-1984 15:59
0000	65	:	Include "Q" command to recalculate cluster quorum
0000	66	:	
0000	67	:	V03-004 TCM0002 Trudy C. Matthews 13-Dec-1983
0000	68	:	New interface to CONSOWNCTY: it now returns the value to
0000	69	:	restore to TXCS in R0 and the value to restore to RXCS
0000	70	:	in R1.
0000	71	:	
0000	72	:	V03-003 TCM0001 Trudy C. Matthews 16-Feb-1983
0000	73	:	Set up to talk to console terminal in a CPU-dependent
0000	74	:	fashion.
0000	75	:	
0000	76	:	V03-002 LMP0024 L. Mark Pilant 23-Apr-1982 13:45
0000	77	:	Correct a problem introduced by some bit shaving.
0000	78	:	
0000	79	:	V03-001 LMP0020 L. Mark Pilant 2-Apr-1982 15:05
0000	80	:	Modify the logic so that only a device that has mount
0000	81	:	verification in progress can be disabled.
0000	82	:	
0000	83	:	V02-001 LMP0010 L. Mark Pilant 9-Feb-1982 9:00
0000	84	:	Strip parity and other garbage from incoming characters.
0000	85	:	--

```

0000 87      .SBTTL  USEFUL MACRO DEFINITIONS
0000 88
0000 89      .MACRO  RJMP  ADDR
0000 90      JMP  ADDR      ;CALLED ROUTINE DOES THE RSB
0000 91      .ENDM  RJMP
0000 92
0000 93      .MACRO  CON MSG MESSAGE,RETURN
0000 94      MOVAB  W^MESSAGE,R2      ;SET MESSAGE ADDRESS
0000 95      .IF  B,      RETURN
0000 96      JSB  (R10)      ;SEND THE MESSAGE
0000 97      .IFF
0000 98      RJMP  (R10)      ;SEND THE MESSAGE
0000 99      .ENDC; B,      RETURN
0000 100     .ENDM  CON_MSG

```



```
0000 102      .SBTTL  SYMBOL DEFINITIONS
0000 103
0000 104      ; MACRO LIBRARY CALLS
0000 105
0000 106      $CLUBDEF
0000 107      $CSBDEF
0000 108      $DCDEF
0000 109      $DDBDEF
0000 110      $DEVDEF
0000 111      $DYNDEF
0000 112      $FKBDEF
0000 113      $IOCDEF
0000 114      $IPLDEF
0000 115      $IRPDEF
0000 116      $PRDEF
0000 117      $UCBDEF
0000 118      $VCBDEF
0000 119
0000 120
0000 121      ; SPECIAL TERMINAL INPUT CHARACTERS
0000 122
00000007 0000 123      IPC_C_BELL=      7      ;BELL (FOR FULL BUFFER & ILLEGAL CHARS)
0000000A 0000 124      IPC_C_LINFEED=    10      ;LINE-FEED (ECHOED AFTER <CR>)
0000000D 0000 125      IPC_C_CARRETN=    13      ;CARRIAGE-RETURN
00000015 0000 126      IPC_C_CNTRLU=     21      ;CONTROL-U (IGNORE THE LINE)
0000001A 0000 127      IPC_C_CNTRLZ=     26      ;CONTROL-Z (EXIT)
00000020 0000 128      IPC_C_SPACE=      32      ;SPACE
0000003A 0000 129      IPC_C_COLON=      58      ;DEVICE NAME TERMINATOR
00000043 0000 130      IPC_C_CHAR_C=     67      ;CHARACTER C (CANCEL MOUNT VERIFICATION)
00000058 0000 131      IPC_C_CHAR_X=     88      ;CHARACTER X (ENTER XDELTA)
0000005F 0000 132      IPC_C_UNDER=     95      ;UNDERSCORE
00000061 0000 133      IPC_C_LOWER=     97      ;LOWER CASE A
0000007F 0000 134      IPC_C_DELETE=    127      ;DELETE A CHARACTER (RUBOUT)
00000051 0000 135      IPC_C_CHAR_Q=     81      ;CHARACTER Q (SET CLUSTER QUORUM)
0000 136
0000 137      ; MISCELLANEOUS CONSTANTS
0000 138
0000000B 0000 139      IPC_C_MAXLINSIZ=    11      ;MAXIMUM INPUT LINE SIZE
0000 140      ; (THIS INCLUDE STORAGE FOR THE <CR>)
0000 141
0000 142      ; LOCAL STORAGE OFFSETS
0000 143
0000 144      $DEFINI IPC
0000 145
0000 146      $DEF      IPC_C_START
0000 147      $DEF      IPC_W_INPSIZE      .BLKW      1      ;INPUT LINE SIZE IN BYTES
0002 148      $DEF      IPC_T_INPLINE      .BLKB      IPC_C_MAXLINSIZ ;STORAGE FOR THE TEXT
000D 149      $DEF      IPC_Q_DEVICE      .BLKQ      1      ;DEVICE DESCRIPTOR
0015 150      $DEF      IPC_B_FLAGS      .BLKB      1      ;USEFUL FLAGS
0016 151      _VIELD      IPC_O,<-      ;DEFINE FLAG BITS
0016 152      <DELETE,,M>,-      ;DELETE CHARACTER SEEN
0016 153      <NEWLINE,,M>,-      ;STARTING A NEW LINE
0016 154      <CNTRLZ,,M>,-      ;CONTROL-Z (EXIT) SEEN
0016 155      >
0016 156      $DEF      IPC_L_OLDGETNXT      .BLKL      1      ;PREVIOUS CHAR INPUT ROUTINE
001A 157
001A 158      $DEF      IPC_L_SAVRXCS      .BLKL      1      ;SAVE ORIGINAL RXCS
```

IPCONTROL  
V04-000

- IPL 12 INTERRUPT PROCESSOR  
SYMBOL DEFINITIONS

J 11

16-SEP-1984 00:26:44 VAX/VMS Macro V04-00  
5-SEP-1984 03:43:51 [SYS.SRC]IPCONTROL.MAR;1

Page 5  
(3)

001E	159	\$DEF	IPC_L_SAVTXCS	.BLKL	1	:SAVE ORIGINAL TXCS
0022	160	\$DEF	IPC_B_PAD	.BLKB	4-<<.-IPC_C_START>&7>	:LONGWORD GRANULARIT
0024	161	\$DEF	IPC_C_LENGTH			:LENGTH OF LOCAL STORAGE
0024	162					
0024	163	\$DEFEND	IPC			



```
0000 165 .SBTTL ERROR MESSAGES AND MISC STRINGS
0000 166
00000000 167 .PSECT WMountVERMSG
0000 168
0000 169 IPC_ILLCOMMAND:
0000 170 .ASCIIZ \illegal command\<13><10>
6D 6D 6F 63 20 6C 61 67 65 6C 6C 49 0000 171 IPC_ILLDEVICE:
00 0A 0D 64 6E 61 000C 172 .ASCIIZ \illegal device\<13><10>
69 76 65 64 20 6C 61 67 65 6C 6C 49 0012 173 IPC_NOMNTVER:
00 0A 0D 65 63 001E 174 .ASCIIZ \Mount verification not in progress\<13><10>
69 66 69 72 65 76 20 74 6E 75 6F 4D 0023
69 20 74 6F 6E 20 6E 6F 69 74 61 63 0023
0A 0D 73 73 65 72 67 6F 72 70 20 6E 002F
00 0047 175 IPC_NOCLUSTER:
20 6E 69 20 74 6F 6E 20 65 64 6F 4E 0048 176 .ASCIIZ \Node not in VAXcluster\<13><10>
0A 0D 72 65 74 73 75 6C 63 58 41 56 0054
00 0060
0061 177
0061 178 IPC_CRLFSTR:
00 0A 0D 0061 179 .BYTE IPC_C_CARRETN,IPC_C_LINFEED,0 ;<CR>,<LF>,0
0064 180
0064 181 IPC_PROMPT:
00 20 3E 43 50 49 0064 182 .ASCIIZ \IPC> \
006A 183
006A 184 .SBTTL CHARACTER VALIDITY TABLES
006A 185
006A 186 IPC_V_LEGAL:
04202000 006A 187 .LONG ^X04202000,- ;<CR>,<^U>,<^Z>
006E 188 ^X07FF0011,- ;<SP>,<$>,<0-9>,<:>
07FF0011 006E 189 ^X87FFFFFFE,- ;<A-Z>,< >
87FFFFFFE 87FFFFFFE 0072 190 ^X87FFFFFFE ;<a-z>,<RUB>
00000000 00000000 00000000 FFFFDB7F 007A 191 IPC_V_ARROW:
007A 192 .LONG ^XFFFFDB7F,0,0,0 ;<BEL>,<LF>,<CR>
008A 193
```

```
008A 195 .SBTTL COMMAND INTERPRETER AND DISPATCHER
008A 196
008A 197 :++
008A 198 : EXESIPCONTROL - IPL 12 (DECIMAL) INTERRUPT HANDLER
008A 199
008A 200 : THIS MODULE IS CALLED WHEN AN IPL 12 INTERRUPT OCCURS. THIS IS NORMALLY
008A 201 : INVOKED FROM THE CONSOLE WHEN THE SYSTEM IS IN SOME DEAD LOCK CONDITION.
008A 202
008A 203 : INPUTS:
008A 204
008A 205 : AN INPUT COMMAND LINE IS ACCEPTED FROM THE CONSOLE FOR ACTION.
008A 206
008A 207 : OUTPUTS:
008A 208
008A 209 : THE APPROPRIATE ACTION IS TAKEN BASED UPON THE CONSOLE INPUT.
008A 210 : THIS IS:
008A 211
008A 212 : 1) CLEAR MOUNT VERIFICATION ON A SPECIFIED DEVICE
008A 213 : 2) TRANSFER CONTROL TO XDELTA
008A 214 : 3) REDUCE CLUSTER QUORUM
008A 215 : 4) EXIT (DISMISS THE IPL 12 INTERRUPT)
008A 216
008A 217 :--
008A 218
00000000 219 .PSECT WIONONPAGED, LONG
0000 220
0000 221 EXESIPCONTROL::
00FF 8F BB 0000 222 PUSHB #^M<R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;SAVE ALL REGS
5E DC AE 9E 0004 223 MOVAB -IPC_C_LENGTH(SP),SP ;MAKE ROOM FOR LOCAL STORAGE
5B 5E D0 0008 224 MOVL SP,RT1 ;SAVE ADDRESS OF LOCAL STORAGE BLOCK
SA 01E0'CF 9E 000B 225 MOVAB W^IPC_OUTPUTZ,R10 ;SET ADDRESS OF .ASCIIZ OUTPUT ROUTINE
00000000'EF 16 0010 226 10$: JSB CON$CONCTY ;SET UP CONSOLE TERMINAL REGISTERS
1E AB 50 D0 0016 227 MOVL R0,IPC_L_SAVTXCS(R11) ;VALUE TO RESTORE TO TXCS
1A AB 51 D0 001A 228 MOVL R1,IPC_L_SAVRXCS(R11) ;VALUE TO RESTORE TO RXCS
56 6B 9E 001E 229 MOVAB IPC_W_INPSIZE(R11),R6 ;SET ADDRESS OF INPUT STRING SIZE
57 02 AB 9E 0021 230 MOVAB IPC_T_INPLINE(R11),R7 ;SET ADDRESS OF INPUT COMMAND BUFFER
58 15 AB 9E 0025 231 MOVAB IPC_B_FLAGS(R11),R8 ;SET ADDRESS OF THE FLAG BYTE
0029 232 CON MSG IPC_PROMPT ;ISSUE A PROMPT
00DF 30 0030 233 BSBB IPC_GETLINE ;GET A COMMAND LINE FROM THE CONSOLE
0B 68 02 E1 0033 234 BBC #IPC_V_CNTRLZ,(R8),15$ ;XFER IF NO CONTROL-Z SEEN
5E 24 AE 9E 0037 235 MOVAB IPC_C_LENGTH(SP),SP ;ELSE CLEAN THE STACK
50 10 003B 236 BSBB IPC_ENABLE ;GO RE-ENABLE THE CONSOLE
00FF 8F BA 003D 237 POPR #^M<R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;RESTORE REGS
02 0041 238 REI ;DISMISS THE INTERRUPT
66 B5 0042 239 15$: TSTW (R6) ;ANYTHING TYPED?
CA 13 0044 240 BEQL 10$ ;XFER IF NOT...RE-ISSUE THE PROMPT
43 8F 67 91 0046 241 20$: CMPB (R7),#IPC_C_CHAR_C ;ELSE CHECK FOR CANCEL
04 12 004A 242 BNEQ 30$ ;XFER IF NOT
52 10 004C 243 BSBB IPC_CANCEL ;ELSE GO CANCEL MOUNT VERIFICATION
C0 11 004E 244 BRB 10$ ;TRY FOR ANOTHER COMMAND
58 8F 67 91 0050 245 30$: CMPB (R7),#IPC_C_CHAR_X ;XFER TO XDELTA?
07 12 0054 246 BNEQ 40$ ;XFER IF NOT
35 10 0056 247 BSBB IPC_ENABLE ;GO RE-ENABLE THE CONSOLE
FFA5' 30 0058 248 BSBB W^INISBRK ;ELSE CAUSE XDELTA BREAKPOINT
B3 11 005B 249 BRB 10$ ;TRY FOR ANOTHER COMMAND
51 8F 67 91 005D 250 40$: CMPB (R7),#IPC_C_CHAR_Q ;REDUCE CLUSTER QUORUM?
20 12 0061 251 BNEQ 50$ ;XFER IF NOT
```

```
50 00000000'GF D0 0063 252      MOVL    G^CLUSGL_CLUB,R0      ;ARE WE IN A CLUSTER?
      09 13 006A 253      BEQL    42$      ;IF NEQ, YES
      50 10 A0 D0 006C 254      MOVL    CLUB$LOCAL(CSB(R0),R0 ;GET LOCAL NODE'S STATUS
09 60 A0 01 E0 0070 255      BBS     #CSB$V-MEMBER,CSB$STATUS(R0) 45$ ;MAKE SURE WE'RE ACTIVE
      92 11 0075 256 42$: CON_MSG IPC_NOCLUSTER      ;REPORT ERROR: NOT IN CLUSTER
      016D 30 007C 257      BRB     10$      ;TRY FOR ANOTHER COMMAND
      8D 11 007E 258 45$: BSBW     IPC_FORK_QUORUM      ;FORK DOWN TO SYNCH TO REDUCE QUORUM
      0081 259      BRB     10$      ;TRY FOR ANOTHER COMMAND
      0083 260
      0083 261 ; ILLEGAL COMMAND GIVEN
      0083 262
      FF83 31 0083 263 50$: CON_MSG IPC_ILLCOMMAND      ;ISSUE AN ERROR MESSAGE
      008A 264      BRW     10$      ;GO TRY FOR ANOTHER
      008D 265
      008D 266 ; LOCAL ROUTINE TO RE-ENABLE THE CONSOLE FOR INTERRUPTS
      008D 267
      008D 268 IPC_ENABLE:
      008D 269      PUSHR   #*M<R0,R1>      ;SAVE REGISTERS
      50 1E AB D0 008F 270      MOVL    IPC_L_SAVTXCS(R11),R0 ;TO BE RESTORED TO TXCS
      51 1A AB D0 0093 271      MOVL    IPC_L_SAVRXCS(R11),R1 ;TO BE RESTORED TO RXCS
00000000'EF 16 0097 272      JSB     CON$RELEASECTY ;RESTORE CONSOLE CTY STATE
      03 BA 009D 273      POPR    #*M<R0,R1> ;RESTORE REGISTERS
      05 009F 274      RSB      ;RETURN TO CALLER
```



```
00A0 276 .SBTTL CANCEL A PENDING MOUNT VERIFICATION
00A0 277
00A0 278 :++
00A0 279 IPC_CANCEL - DISABLE MOUNT VERIFICATION IF IN PROGRESS
00A0 280
00A0 281 THIS ROUTINE DISABLES MOUNT VERIFICATION IF IT IS IN PROGRESS. THIS IS
00A0 282 DEFINED AS BEING IN THE STATE IN WHICH MOUNT VERIFICATION IS ENABLED
00A0 283 AND EITHER THE DEVICE UCB HAS THE MOUNT VERIFICATION IN PROGRESS STATUS
00A0 284 BIT SET OR THE IRP AT THE HEAD OF THE DEVICE IRP QUEUE HAS THE MOUNT
00A0 285 VERIFICATION STATUS BIT SET.
00A0 286
00A0 287 INPUTS:
00A0 288
00A0 289 R7 - ADDRESS OF THE CONSOLE INPUT LINE BUFFER
00A0 290
00A0 291 OUTPUTS:
00A0 292
00A0 293 1) THE MOUNT VERIFICATION BIT IS CLEARED, OR
00A0 294 2) A WARNING MESSAGE IS ISSUED
00A0 295
00A0 296 :--
00A0 297
00A0 298 IPC_CANCEL:
50 57 D0 00A0 299 MOVL R7,R0 ;SET ADDRESS OF COMMAND LINE
80 95 00A3 300 TSTB (R0)+ ;SKIP OVER COMMAND CHARACTER
20 80 91 00A5 301 CMPB (R0)+,#*A/ / ;CHECK FOR CORRECT SEPARATOR
07 13 00A8 302 BEQL 10$ ;XFER IF CORRECT
00AA 303 CON MSG IPC_ILLCOMMAND,RETURN ;ISSUE MESSAGE AND RETURN
SF 8F 80 91 00B1 304 10$: CMPB (R0)+,#IPC_C_UNDER ;ELSE CHECK FOR AN UNDERSCORE
FA 13 00B5 305 BEQL 10$ ;XFER IF THERE IS ONE
50 D7 00B7 306 DECL R0 ;ELSE BACK UP OVER NON-UNDERSCORE
0D AB D4 00B9 307 CLRL IPC_Q_DEVICE(R11) ;RESET DEVICE NAME SIZE
11 AB 50 D0 00BC 308 MOVL R0,IPC_Q_DEVICE+4(R11) ;SET STARTING ADDRESS OF DEVICE NAME
0D 60 91 00C0 309 20$: CMPB (R0),#IPC_C_CARRETN ;END OF THE STRING?
0A 13 00C3 310 BEQL 30$ ;XFER IF SO
3A 80 91 00C5 311 CMPB (R0)+,#IPC_C_COLON ;ALTERNATE TERMINATOR?
05 13 00C8 312 BEQL 30$ ;XFER IF SO
0D AB B6 00CA 313 INCW IPC_Q_DEVICE(R11) ;ELSE COUNT ONE MORE CHARACTER
FI 11 00CD 314 BRB 20$ ;GO TRY THE NEXT CHARACTER
00CF 315
00CF 316 ; FIND GENERIC DEVICE NAME (WITH CONTROLLER) AND UNIT NUMBER (IN BINARY)
00CF 317
7E 5A 7D 00CF 318 30$: MOVQ R10,-(SP) ;SAVE R10 & R11 THROUGH SEARCH CALL
58 0D AB 7D 00D2 319 MOVQ IPC_Q_DEVICE(R11),R8 ;GET DEVICE NAME DESCRIPTOR
5A 00000000 00000041 8F 7D 00D6 320 MOVQ #IOC$M_PHY!IOC$M_ANY,R10 ;PHYSICAL SEARCH, NO AVAILABILITY CHECKS
00E1 321 ;LKS(B(R11)) = 0
FF1C' 30 00E1 322 BSBW IOC$PARSDEVNAM ;PARSE THE SUPPLIED DEVICE NAME
03 50 E9 00E4 323 BLBC R0,40$ ;BRANCH OUT ON ERROR
FF16' 30 00E7 324 BSBW IOC$SEARCHINT ;SEARCH THE I/O DATABASE
5A 8E 7D 00EA 325 40$: MOVQ (SP)+,R10 ;RESTORE R10 & R11
14 50 E9 00ED 326 BLBC R0,70$ ;OUT ON ERROR
01 40 A5 91 00F0 327 CMPB UCBSB_DEVCLASS(R5),#DCS_DISK ;IS DEVICE A DISK?
0E 12 00F4 328 BNEQ 70$ ;BRANCH IF NOT
10 64 A5 0E E1 00F6 329 50$: BBC #UCBSV_MNTVERIP,UCBSW_STS(R5),80$ ;BRANCH IF NOT IN PROGRESS
52 34 A5 D0 00FB 330 60$: MOVL UCBSL_VCB(R5),R2 ;GET VCB ADDRESS
53 A2 04 8A 00FF 331 BICB2 #1@VCBSV_MOUNTVER,VCBSB_STATUS2(R2) ;ZAP MOUNT VERIFICATION
05 0103 332 RSB ;RETURN NOW
```

IPCONTROL  
V04-000

- IPL 12 INTERRUPT PROCESSOR B 12  
CANCEL A PENDING MOUNT VERIFICATION

16-SEP-1984 00:26:44 VAX/VMS Macro V04-00  
5-SEP-1984 03:43:51 [SYS.SRC]IPCONTROL.MAR;1

Page 10  
(6)

0104	333		
0104	334	70\$:	CON_MSG IPC_ILLDEVICE,RETURN ;ELSE ISSUE MESSAGE AND RETURN
010B	335	80\$:	CON_MSG IPC_NOMNTVER,RETURN ;ELSE ISSUE ERROR AND RETURN

```
0112 337 .SBTTL COMMAND LINE INPUT ROUTINE
0112 338
0112 339 :++
0112 340
0112 341 IPC_GETLINE - GET A LINE OF INPUT FROM THE CONSOLE
0112 342
0112 343 THIS ROUTINE INPUTS A LINE FROM THE CONSOLE. IT IGNORES LEADING SPACES,
0112 344 COMPRESSES MULTIPLE SPACES TO A SINGLE SPACE, CONVERTS LOWER CASE TO UPPER
0112 345 CASE, HANDLES CHARACTER (RUBOUT) AND LINE (CONTROL-U) DELETION, AND CHECKS
0112 346 THE LEGALITY OF ANY CHARACTER RECEIVED.
0112 347
0112 348 INPUTS:
0112 349
0112 350 R6 - ADDRESS OF THE INPUT LINE SIZE STORAGE
0112 351 R7 - ADDRESS OF THE INPUT LINE BUFFER
0112 352 R11 - ADDRESS OF THE LOCAL STORAGE AREA
0112 353
0112 354 OUTPUTS:
0112 355
0112 356 IPC_W_INPSIZE(R11) - SIZE OF USER INPUT LINE AFTER COMPRESSION
0112 357 IPC_T_INPLINE(R11) - STORAGE FOR THE CONSOLE INPUT
0112 358
0112 359 R0,R1,R2, AND R8 ARE DESTROYED
0112 360
0112 361 :--
0112 362
0112 363 IPC_GETLINE:
0112 364 CLR R8 ;RESET ALL FLAGS
0114 365 MOVL R7,R2 ;COPY BUFFER ADDRESS
0117 366 10$: CLR R6 ;RESET INPUT LINE SIZE
0119 367 BISB2 #IPC_M_NEWLINE,(R8) ;NEW LINE ONLY USEFUL FLAG
011C 368 20$: MFPR #PR$ RXCS,R0 ;GET DEVICE STATUS
011F 369 BBC #7,R0,20$ ;XFER IF NOT READY
0123 370 MFPR #PR$ RXDB,R0 ;ELSE GET THE CHARACTER
0126 371 BITW #^XFF00,R0 ;ERROR OF NOT CONSOLE INPUT?
012B 372 BNEQ 20$ ;YES, IGNORE
012D 373 BICB2 #^X80,R0 ;ZAP PARITY BIT IF ANY
0131 374 BBC R0,#IPC_V_LEGAL,70$ ;CHECK LEGALITY
0137 375 CMPB R0,#IPC_C_DELETE ;CHECK FOR CHARACTER DELETE
013B 376 BNEQ 35$ ;XFER IF NOT
013D 377 BBS #IPC_V_NEWLINE,(R8),20$ ;IGNORE IF NOTHING THERE
0141 378 BBSS #IPC_V_DELETE,(R8),30$ ;XFER IF HERE ONCE ALREADY
0145 379 MOVZBL #A/T/,R0 ;SET DELIMITER
0149 380 BSBB IPC_OUTCHR ;TYPE IT OUT
014B 381 30$: DECB (R6) ;ANYTHING TO DELETE?
014D 382 BLSS 10$ ;XFER IF NOT...SET NEW LINE
014F 383 MOVZBL -(R2),R0 ;GET CHARACTER
0152 384 BSBB IPC_OUTCHR ;TYPE IT OUT
0154 385 BRB 20$ ;GET NEXT CHARACTER FROM THE CONSOLE
0156 386 35$: CMPB R0,#IPC_C_CNTRLZ ;END OF THE LINE?
0159 387 BNEQ 40$ ;XFER IF NOT
015B 388 BISB2 #IPC_M_CNTRLZ,(R8) ;ELSE NOTE IT
015E 389 BRB 45$ ;GO FINISH UP
0160 390 40$: CMPB R0,#IPC_C_CNTRLU ;DELETE ENTIRE LINE?
0163 391 BNEQ 50$ ;XFER IF NOT
0165 392 45$: BSBB IPC_OUTCHR ;ECHO ^U
0167 393 BSBB IPC_CRLF ;NEW LINE
```



	66	B4	0169	394	CLRW	(R6)	;NOTHING THERE (FORCE NEW PROMPT)
		05	016B	395	RSB		;RETURN TO MAIN ROUTINE
20	50	91	016C	396	50\$: CMPB	RO,#IPC_C_SPACE	;SPACE CHARACTER?
	0A	12	016F	397	BNEQ	60\$	;XFER IF NOT
A7 68	01	E0	0171	398	BBS	#IPC_V_NEWLINE,(R8),20\$	;XFER IF LEADING SPACE
20 FF	A2	91	0175	399	CMPB	-1(R2),#IPC_C_SPACE	;PREVIOUS CHARACTER A SPACE?
	A1	13	0179	400	BEQL	20\$	;IGNORE IF SO
0B	66	B1	017B	401	60\$: CMPW	(R6),#IPC_C_MAXLINSIZ	;TOO MANY CHARACTERS?
	07	19	017E	402	BLSS	80\$	;XFER IF NOT
50	07	9A	0180	403	70\$: MOVZBL	#IPC_C_BELL,RO	;ELSE SET NEW CHARACTER
	31	10	0183	404	BSBB	IPC_OUTCHR	;LET USER KNOW
	95	11	0185	405	BRB	20\$	;GO GET THE NEXT CHARACTER
61 8F	50	91	0187	406	80\$: CMPB	RO,#IPC_C_LOWER	;LOWER CASE CHARACTER?
	03	19	018B	407	BLSS	90\$	;XFER IF NOT
50	20	8A	018D	408	BICB2	#^X20,RO	;ELSE CONVERT TO UPPER CASE
0A 68	00	E1	0190	409	90\$: BBC	#IPC_V_DELETE,(R8),100\$	;FINISH OUT DELETE STRING?
	01	BB	0194	410	PUSHR	#^M<R0\$	;YES, SAVE CURRENT CHARACTER
50 5C	8F	9A	0196	411	MOVZBL	#^A/\/,RO	;SET TERMINATOR
	1A	10	019A	412	BSBB	IPC_OUTCHR	;TYPE IT OUT
	01	BA	019C	413	POPR	#^M<R0>	;RESTORE ORIGINAL CHARACTER
82	50	90	019E	414	100\$: MOVB	RO,(R2)+	;SAVE THE CHARACTER
68	02	8A	01A1	415	BICB2	#IPC_M_NEWLINE,(R8)	;NO LONGER EMPTY LINE
	10	10	01A4	416	BSBB	IPC_OUTCHR	;TYPE OUT THE CHARACTER
0D	50	91	01A6	417	CMPB	RO,#IPC_C_CARRETN	;END OF THE LINE?
	06	12	01A9	418	BNEQ	110\$	;XFER IF MORE TO COME
50	0A	9A	01AB	419	MOVZBL	#IPC_C_LINFEED,RO	;ELSE FOLLOW WITH A <LF>
	06	10	01AE	420	BSBB	IPC_OUTCHR	
		05	01B0	421	RSB		;RETURN TO CALLER
	66	B6	01B1	422	110\$: INCW	(R6)	;ONE MORE CHARACTER
FF66	31	01B3	423	BRW	20\$		;GO GET ANOTHER

```
01B6 425 .SBTTL MISCELLANEOUS CONSOLE OUTPUT ROUTINES
01B6 426
01B6 427 :++
01B6 428
01B6 429 IPC_OUTCHR - OUTPUT A SINGLE CHARACTER
01B6 430
01B6 431 THIS ROUTINE OUTPUTS A SINGLE CHARACTER TO THE CONSOLE. IF IS A SPECIAL
01B6 432 ACTION CONTROL CHARACTER (CR, BELL, OR LF) THEN IT IS TYPED DIRECTLY.
01B6 433 OTHERWISE IT IS PRECEDED BY AN UP-ARROW (^).
01B6 434
01B6 435 INPUTS:
01B6 436
01B6 437 RO - CHARACTER TO TYPE
01B6 438
01B6 439 OUTPUTS:
01B6 440
01B6 441 NONE
01B6 442
01B6 443 R1 DESTROYED
01B6 444
01B6 445 :--
01B6 446
01B6 447 IPC_OUTCHR:
01B6 448 PUSHL RO ;SAVE THE CHARACTER
01B6 449 BBC (SP),W^IPC_V_ARROW,20$ ;PRECEED WITH AN UP ARROW?
01B6 450 10$: MFPR #PRS_TXCS,R1 ;YES, GET CONSOLE STATUS
01B6 451 BBC #7,RT,10$ ;XFER IF NOT READY
01B6 452 MTPR #^A\^,\, #PRS_TXDB ;ELSE TYPE THE UP ARROW
01B6 453 BISB2 #^X40,(SP) ;CONVERT TO A PRINTABLE CHARACTER
01B6 454 20$: MFPR #PRS_TXCS,R1 ;GET DEVICE STATUS
01B6 455 BBC #7,RT,20$ ;XFER IF NOT READY
01B6 456 MTPR (SP)+, #PRS_TXDB ;ELSE OUTPUT THE CHARACTER
01B6 457 RSB ;RETURN TO CALLER
01B6 458
01B6 459 :--
01B6 460
01B6 461 IPC_CRLF - OUTPUT A <CR,LF> SEQUENCE
01B6 462
01B6 463 THIS ROUTINE OUTPUTS A <CR,LF> SEQUENCE TO THE CONSOLE. IT FALLS THROUGH
01B6 464 TO THE STRING OUTPUT ROUTINE TO AVOID THE UNNEEDED SUBROUTINE CALL AND RETURN.
01B6 465 (THE RETURN IS DONE BY THE STRING OUTPUT ROUTINE.)
01B6 466
01B6 467 INPUTS:
01B6 468
01B6 469 NONE
01B6 470
01B6 471 OUTPUTS:
01B6 472
01B6 473 NONE
01B6 474
01B6 475 :--
01B6 476
01B6 477 IPC_CRLF:
01B6 478 MOVAB W^IPC_CRLFSTR,R2 ;SET MESSAGE ADDRESS
01E0 479 ;FALL THROUGH INTO STRING ROUTINE
01E0 480
01E0 481 :++
```

12 007A'CF 50 DD 01B6 448  
51 6E E1 01B8 449  
F9 51 22 DB 01BE 450 10\$:  
23 0000005E 07 E1 01C1 451  
6E 40 8F DA 01C5 452  
51 22 DB 01D0 453  
F9 51 07 E1 01D3 454 20\$:  
23 8E DA 01D7 455  
05 01DA 456  
01DB 457  
01DB 458  
01DB 459  
01DB 460  
01DB 461  
01DB 462  
01DB 463  
01DB 464  
01DB 465  
01DB 466  
01DB 467  
01DB 468  
01DB 469  
01DB 470  
01DB 471  
01DB 472  
01DB 473  
01DB 474  
01DB 475  
01DB 476  
52 0061'CF 9E 01DB 477  
01E0 478  
01E0 479  
01E0 480  
01E0 481

```
01E0 482 :  
01E0 483 : IPC_OUTPUTZ - TYPE OUT AN .ASCII STRING ON THE CONSOLE  
01E0 484 :  
01E0 485 : THIS ROUTINE TYPES OUT A ZERO TERMINATED STRING ON THE CONSOLE.  
01E0 486 :  
01E0 487 : INPUTS:  
01E0 488 :  
01E0 489 :     R2 - ADDRESS OF THE STRING TO TYPE  
01E0 490 :  
01E0 491 : OUTPUTS:  
01E0 492 :  
01E0 493 :     NONE  
01E0 494 :  
01E0 495 : --  
01E0 496 :  
50 82 9A 01E0 497 IPC_OUTPUTZ:  
04 13 01E0 498     MOVZBL (R2)+,R0           ;GET A CHARACTER  
CF 10 01E3 499     BEQL 10$           ;XFER WHEN DONE  
F7 11 01E5 500     BSBB IPC_OUTCHR    ;ELSE TYPE OUT THE CHARACTER  
05 01E7 501     BRB IPC_OUTPUTZ      ;GO GET THE NEXT  
01E9 502 10$:     RSB                 ;RETURN WHEN DONE  
01EA 503 :  
01EA 504 :  
01EA 505 : ++  
01EA 506 :  
01EA 507 : IPC_FORK_QUORUM  
01EA 508 :  
01EA 509 : THIS ROUTINE QUEUES A FORK TO IPL$ SYNCH IN ORDER TO SYNCHRONIZE WITH  
01EA 510 : THE CLUSTER CONNECTION MANAGER. THE FORK ROUTINE REQUESTS A RECALCULATION  
01EA 511 : OF DYNAMIC QUORUM BASED ON THE CURRENT CLUSTER CONFIGURATION. THIS IS DONE  
01EA 512 : WHEN THE CLUSTER IS HUNG DUE TO LACK OF QUORUM BECAUSE A CRITICAL NODE HAS  
01EA 513 : CRASHED AND CANNOT BE REBOOTED.  
01EA 514 :  
01EA 515 : INPUTS:  
01EA 516 :  
01EA 517 :     NONE  
01EA 518 :  
01EA 519 : OUTPUTS:  
01EA 520 :  
01EA 521 :     NONE  
01EA 522 :  
01EA 523 : --  
01EA 524 :     ASSUME FKB$B_TYPE EQ FKB$W_SIZE+2  
01EA 525 :     ASSUME FKB$B_FIPL EQ FKB$W_SIZE+3  
01EA 526 FKB_INIT: ;INITIALIZATION FOR FORK BLOCK  
0018 01EA 527     .WORD FKB$K_LENGTH      ; FKB$W_SIZE  
08 01EC 528     .BYTE DYN$C_FK      ; FKB$B_TYPE  
08 01ED 529     .BYTE IPL$_SYNCH    ; FKB$B_FIPL  
01EE 530 :  
01EE 531 : IPC_FORK_QUORUM:  
01EE 532 :  
01EE 533 :     ALLOCATE A FORK BLOCK FROM NONPAGED POOL. INITIALIZE IT.  
01EE 534 :  
51 18 D0 01EE 535     MOVL #FKB$K_LENGTH,R1      ;LENGTH OF FORK BLOCK  
00000000'GF 16 01F1 536     JSB G^EX$ALONONPAGED    ;GET THE BLOCK  
26 50 E9 01F7 537     BLBC R0,10$      ;IF LBC, DIDN'T GET IT.  
55 52 D0 01FA 538     MOVL R2,R5          ;PUT IT WHERE IT BELONGS
```



```
08 A5 EA AF D0 01FD 539      MOVL    FKB_INIT,FKBSW_SIZE(R5)      ;INIT SIZE, TYPE, FIPL
                   0202 540      :
                   0202 541      : FORK DOWN TO IPL$_SYNCH ...
                   0202 542      :
                   0202 543      : FORK
                   0208 544      :
                   0208 545      :
                   0208 546      : AT THIS STAGE, WE ARE EXECUTING IN THE FORK AT IPL SYNCH.
                   0208 547      :
                   55 DD 0208 548 5$: PUSHL    R5                  ;SAVE FORK BLOCK POINTER
                   51 D4 020A 549      CLRL     R1                  ;QUORUM=0 MEANS YOU FIGURE IT OUT.
00000000'GF 16 020C 550      JSB      G^CNX$CHANGE_QUORUM        ;CALL CONNECTION MANAGER
                   55 8ED0 0212 551      POPL     R5                  ;RESTORE FORK BLOCK POINTER
                   0213 552      :
                   0213 553      : IF WE GET ANY ERROR RETURNED, WAIT ONE SECOND AND RETRY.
                   0213 554      : CONNECTION MANAGER CANNOT ALWAYS DO THIS THE FIRST TIME IT TRIES.
                   0213 555      :
                   08 50 E8 0213 556      BLBS     R0,10$           ;LBS MEANS IT WORKED
                   E8 11 0218 557      FORK_WAIT
                   0220 558      BRB      5$                       ;WAIT ONE SECOND
                   0220 559      :
                   50 55 D0 0220 561 10$: MOVL     R5,R0            ;TRY AGAIN.
00000000'GF 16 0223 562      JSB      G^EXE$DEANONPAGED          ;EVENTUALLY IT WILL WORK (SEZ HERE)
                   05 0229 563      RSB
                   0229 563      :
```

IPCONTROL  
V04-000

- IPL 12 INTERRUPT PROCESSOR H 12  
MISCELLANEOUS CONSOLE OUTPUT ROUTINES

16-SEP-1984 00:26:44 VAX/VMS Macro V04-00  
5-SEP-1984 03:43:51 [SYS.SRC]IPCONTROL.MAR;1

Page 16  
(9)

022A 565 .END

IPCONTROL  
Symbol table

- IPL 12 INTERRUPT PROCESSOR

I 12

16-SEP-1984 00:26:44 VAX/VMS Macro V04-00  
5-SEP-1984 03:43:51 [SYS.SRC]IPCONTROL.MAR;1

Page 17  
(9)

CLUSGL CLUB	*****	X	03
CLUBSL-LOCAL CSB	= 00000010		
CNX\$CHANGE-QUORUM	*****	X	03
CON\$OWNCTY	*****	X	03
CON\$RELEASECTY	*****	X	03
CSBSL STATUS	= 00000060		
CSBSV-MEMBER	= 00000001		
DC\$ DISK	= 00000001		
DYN\$C FRK	= 00000008		
EXESACNONPAGED	*****	X	03
EXESDEANONPAGED	*****	X	03
EXESFORK	*****	X	03
EXESFORK WAIT	*****	X	03
EXESIPCONTROL	00000000	RG	03
FKBSB_FIPL	= 0000000B		
FKBSB-TYPE	= 0000000A		
FKBSK-LENGTH	= 00000018		
FKBSW-SIZE	= 00000008		
FKB INIT	000001EA	R	03
INISBRK	*****	X	03
IOCSM-ANY	= 00000040		
IOCSM-PHY	= 00000001		
IOCSPARSDEVNAM	*****	X	03
IOCSSEARCHINT	*****	X	03
IPC_B_FLAGS	00000015		
IPC_B-PAD	00000022		
IPC-CANCEL	000000A0	R	03
IPC-CRLF	000001DB	R	03
IPC-CRLFSTR	00000061	R	02
IPC-C_BELL	= 00000007		
IPC-C-CARRETN	= 0000000D		
IPC-C-CHAR-C	= 00000043		
IPC-C-CHAR-Q	= 00000051		
IPC-C-CHAR-X	= 00000058		
IPC-C-CNTRCU	= 00000015		
IPC-C-CNTRLZ	= 0000001A		
IPC-C-COLON	= 0000003A		
IPC-C-DELETE	= 0000007F		
IPC-C-LENGTH	= 00000024		
IPC-C-LINFEEED	= 0000000A		
IPC-C-LOWERA	= 00000061		
IPC-C-MAXLINSIZ	= 0000000B		
IPC-C-SPACE	= 00000020		
IPC-C-START	= 00000000		
IPC-C-UNDER	= 0000005F		
IPC-ENABLE	0000008D	R	03
IPC-FORK-QUORUM	000001EE	R	03
IPC-GETLINE	00000112	R	03
IPC-ILLCOMMAND	00000000	R	02
IPC-ILLDEVICE	00000012	R	02
IPC-L-OLDGETNXT	00000016		
IPC-L-SAVRXCS	0000001A		
IPC-L-SAVTXCS	0000001E		
IPC-M-CNTRLZ	= 00000004		
IPC-M-NEWLINE	= 00000002		
IPC-NOCLUSTER	00000048	R	02
IPC-NOMNTVER	00000023	R	02

IPC-OUTCHR	000001B6	R	03
IPC-OUTPUTZ	000001E0	R	03
IPC-PROMPT	00000064	R	02
IPC-Q-DEVICE	0000000D		
IPC-T-INPLINE	00000002		
IPC-V-ARROW	0000007A	R	02
IPC-V-CNTRLZ	= 00000002		
IPC-V-DELETE	= 00000000		
IPC-V-LEGAL	0000006A	R	02
IPC-V-NEWLINE	= 00000001		
IPC-W-INPSIZE	00000000		
IPL\$ SYNCH	= 00000008		
PR\$-RXCS	= 00000020		
PR\$-RXDB	= 00000021		
PR\$-TXCS	= 00000022		
PR\$-TXDB	= 00000023		
SIZ...	= 00000001		
UCBSB-DEVCLASS	= 00000040		
UCBSL-VCB	= 00000034		
UCBSV-MNTVERIP	= 0000000E		
UCBSW-STS	= 00000064		
VCBSB-STATUS2	= 00000053		
VCBSV-MOUNTVER	= 00000002		



+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000024 ( 36.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
WMOUNTVERMSG	0000008A ( 138.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
WIONONPAGED	0000022A ( 554.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:01.07
Command processing	105	00:00:00.56	00:00:03.90
Pass 1	391	00:00:14.00	00:00:43.52
Symbol table sort	0	00:00:02.27	00:00:08.11
Pass 2	112	00:00:02.72	00:00:10.66
Symbol table output	9	00:00:00.10	00:00:00.23
Psect synopsis output	3	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	651	00:00:19.73	00:01:07.67

The working set limit was 1350 pages.

81107 bytes (159 pages) of virtual memory were used to buffer the intermediate code.

There were 80 pages of symbol table space allocated to hold 1539 non-local and 34 local symbols.

565 source lines were read in Pass 1, producing 16 object records in Pass 2.

27 pages of virtual memory were used to define 26 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	12
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	20

1604 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:IPCONTROL/OBJ=OBJ\$:IPCONTROL MSRC\$:IPCONTROL/UPDATE=(ENH\$:IPCONTROL)+EXECMLS/LIB



0376

AH-BT13A-SE  
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY